Ｏｂｊｅｃｔ－Ｚのための要求定義と形式的プログラム導出

3B－8

宮崎比呂志
情報処理振興事業協会(富士通より出向)

1．はじめに

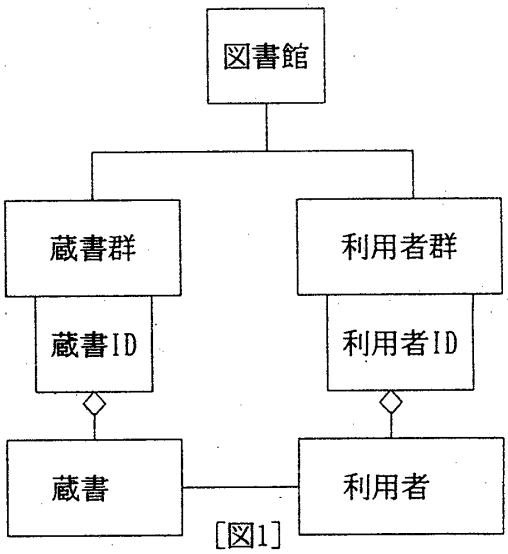
通常，プログラムの仕様記述は，構造化プログラムなどの図的な表現によって記述されるか，あるいは，自然言語で記述されるのが一般的である．図的なプログラム仕様などでは，制御の流れなどに依存したものとなる．したがって，仕様の意味論の抽象化がされておらず可読性の低いものとなる．また，自然言語による仕様では，自然言語による仕様では，自然言語の曖昧性などにより一意的な解釈ができずに，仕様本来の目的を果たさない場合がある．このような問題から，仕様やプログラムの意味論の抽象化が可能であり，曖昧さのない仕様記述方法が必要となる．この問題を解決するのが，論理式などで表す形式的仕様記述法である．一方，形式的仕様記述言語であるObject-Z[*1]などは，実行不可能なものである．したがって，プログラムに結びつけるためには，仕様の内容を解釈して，それ同値と思われるプログラムに変換せざるを得ない．それゆえ，この変換の際に誤ったアルゴリズムを導く可能性もあり，厳密に仕様記述した意味が失われる場合がある．

本稿では，Object-Z仕様記述のための分析／設計法と，仕様から正しいプログラムに変換するための方法について，図書館問題[*2]を例題として述べる．

2.Object-Zのための概念モデル

Object-Zはオジェクト指向の形式的仕様記述言語である．したがって，Object-Z仕様を記述するための分析／設計技法はOMT [*3]を用いる．

OMT では，システムの静的にオブジェクトの構造を定義するオブジェクトモデル，制御の流れを定義するダイナミックモデル，オブジェクトに対する操作を定義するファンクショナルモデルを記述するようになっている．ここでは，ダイナミックモデルとファンクショナルモデルのデータフロー図を設計することは省略した．図1 はOMT により分析/ 設計したオブジェクトモデルの概略図である．



[図1]

Requirement Definition and Formal Delivation for Object-Z

Hiroshi Miyazaki
Information-technology Promotion Agency(also with FUJITSU)
1-38 Shibakoen 3-choume, MINATO-ku, TOKYO 105, JAPAN

3.Object-Z仕様

オブジェクト指向の分析／設計法を用いることにより，オブジェクトの構造を決定することができた．したがって，このOMT モデルを基にして，Object-Zの仕様を作成する．Object-Zの仕様を図2 に示す．ただし，Object-Z の仕様は紙面の都合上，蔵書クラスと蔵書群クラスの貸出操作である．

蔵書クラムの貸出

```
貸出
Δ(status, borrower, bordate)
uc?:利用者ID
d?: 日付
─────────────────
status  = "在庫中"
status' = "貸出中"
borrower' = uc?
bordate'  = d?
```

蔵書群クラスの貸出

```
貸出
Δ(copies)
cc?:蔵書ID
uc?:利用者ID
d?: 日付
─────────────────
cc? ∈ dom(copies)
let c==copies(cc?) ・∃c': 蔵書 | c.貸出・
      copies' = copies ⊕ {cc? ↦c'}
```

[図2]

3.プログラム導出

この仕様からプログラムを導出する．Object-Zの仕様から，非形式的にプログラムに変換したのでは，誤ったプログラムを導く可能性もあり，厳密に仕様記述をした意味が失われる．したがって，Object-Zの仕様と同値なプログラムになるように，形式的プログラム導出法を用いることにする．形式的プログラム導出法としては，Dijkstra-Gries [*4, *5] 流の導出法を用いる．Dijkstra-Gries流の導出法は，プログラム仕様を事前条件(pre- conditino)と事後条件(post-condition)として一階述語論理で記述するものである．プログラムは仕様を代数的な操作で変形することによって導出することが可能である．Object-Z仕様からGries の仕様への変換は，集合で表した仕様を配列に関する一階述語論理に変換することによって求められる．Object-Z仕様からGries 流の仕様に変換した時の同値性に関してはZ の "具体化(refinement)の証明" を用いることによって保証した．導出するプログラムはAda とした．

5.まとめと課題

本稿では，Object-Z仕様記述をするための問題分析／設計法と，Object-Zからのプログラムコードの生成までを行った．このような方法を採ることにより，クラス分けのための手法，及び形式的なプログラム導出を行うことができた．課題としては，このようにクラス分けの正当性を検証することのできる手段を考慮する必要がある．

6.参考文献

[*1] D. Carrington, D. Duke, P. King, G. Rose, G. Smith, Object-Z:An Object-Oriented Extension to Z, In Formal Description Techniques (FORTE'89). North Holland, 1990.

[*2] Problem Set for the Fourth International Workshop on Software Specification and Design 1987 MONTEREY, CALIFORNIA, USA

[*3] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen, Object-Oriendted Modeling and Design. Prentice Hall, 1991
羽生田栄一監訳「ｵﾌﾞｼﾞｪｸﾄ 指向方法論OMT 」ﾄｯﾊﾟ

[*4] E. W. Dijkstra, Dicsioline of Programming. Prentice Hall, 1976
浦昭二，土居範久，原田賢一共訳「ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ原論」ｻｲｴﾝｽ 社

[*5] D. Gries, The Science of Programming. Springer-Verlag, 1981
筧　捷彦訳「ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ の科学」培風館